IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

APPEAL BRIEF – 37 C.F.R § 1.192

U.S. Patent Application 10/711,808 entitled:

"TRANSIENT RANGE VERSIONING BASED ON REDIRECTION"

Real Party in Interest: International Business Machines Corporation

Related Appeals and Interferences:

None

Status of Claims:

Claims 1-25 were previously canceled.

Claims 26-49 are pending.

Claims 26-32, 34-45 and 47-49 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Ganesh et al. (U.S. Patent 6,957,236), hereafter "Ganesh" in view of Odom et al. (U.S. Patent 6,516,320), hereafter "Odom", and further in view of Najork et al. (U.S. Patent 7,007,027), hereafter "Najork".

Claims 33 and 46 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Ganesh in view of Odom, and further in view of Najork, and further in view of Chang et al. (U.S. 6,584,459), hereafter "Chang".

Claims 26-49 are hereby appealed.

Status of Amendments:

No Amendments were filed after the Final Rejection dated 3/03/2008.

Summary of Claimed Subject Matter:

(NOTE: All citations are made from the original specification, including the figures.)

According to pending **claim 26**, the present invention provides for a computer-based method to version a node range and locate a versioned node range in a storage architecture managing node ranges, said computer-based method implemented in computer readable program code stored in computer memory, said computer-based method comprising the steps of: (a) receiving a node modification request for a node range from a database system (**see paragraphs [0015], [0051], [0052] of application-as-filed**); (b) versioning said node range by copying, to a storage, a node range to which said node modification request is to be made and labeling said copied node range with an identifier (**see paragraphs [0015], [0051], [0052] of application-as-filed**); (c) locating said labeled node range via said identifier and a hash on said node range (**see paragraphs [0015], [0051], [0052] of application-as-filed**); and (d) outputting said located labeled node range (**see paragraphs [0015], [0051], [0052] of application-as-filed**).

According to pending **claim 34**, the present invention provides for a computer-based method to version a node range and to locate a versioned node range in a storage architecture managing node ranges via a node id range index, said each node assigned a node id value and a set of nodes forming a node range, each entry in said node id range index pointing to a node range and its range identifier, RID, said computer-based method implemented in computer readable program code stored in computer memory, said method comprising the steps of: (a) receiving a node modification request for a range (**see paragraphs [0016], [0026], [0051], [0052] of application-as-filed**); (b) versioning said range associated with said node modification request by shadowing nodes in said range to a Version Hash Table based on RID and assigning a time identifier to copies of said range (**see paragraphs [0016], [0026], [0051], [0052] of**

application-as-filed); (c) locating a node in said shadowed range via said time identifier and RIDs (**see paragraphs [0016], [0026], [0051], [0052] of application-as-filed**); and (d) outputting said located node range (**see paragraphs [0016], [0026], [0051], [0052] of application-as-filed**).

According to pending **claim 48**, the present invention provides for an article of manufacture comprising computer readable program code implementing a method to version a node range and to locate said versioned node in a storage architecture that manages node ranges via a node id range index, said each node assigned a node id value and a set of nodes forming a node range, each entry in said node id range index pointing to a node range and its range identifier, RID, said method comprising: (a) computer readable program code identifying a node modification request for a range (**see paragraphs [0016], [0017], [0028], [0051], [0052] of application-as-filed**); (b) computer readable program code versioning said range associated with said node modification request by shadowing nodes in said range to a Version Hash Table based on RID and assigning a time identifier to copies of said range (**see paragraphs [0016], [0017], [0028], [0051], [0052] of application-as-filed**); (c) computer readable program code locating a node in said shadowed range via said time identifier and RIDs (**see paragraphs [0016], [0017], [0028], [0051], [0052] of application-as-filed**); and (d) computer readable program code outputting said located node range (**see paragraphs [0016], [0017], [0028], [0051], [0052] of application-as-filed**).

According to pending **claim 49**, the present invention provides for an article of manufacture comprising computer readable program code implementing a method to version a node range and to locate a versioned node rage in a storage architecture that manages node ranges, said method comprising: (a) computer readable program code identifying a request for node modification from a database system (**see paragraphs [0015], [0017], [0026], [0051], [0052] of application-as-filed**); (b) computer readable program code copying, to a storage, a node range to which said node modification request is to be made (**see paragraphs [0015], [0017], [0026], [0051], [0052] of application-as-filed**); (c) computer readable program code labeling said copied node range with an identifier (**see paragraphs [0015], [0017], [0026], [0051], [0052] of application-as-filed**); and (d) computer readable program code locating said labeled node range via said identifier and a hash on said node range (**see paragraphs [0015], [0017], [0026], [0051], [0052] of application-as-filed**); and (e) computer readable program code outputting said located labeled node range (**see paragraphs [0015], [0017], [0026], [0051], [0052] of application-as-filed**).

Grounds of Rejection to be Reviewed on Appeal:

Claims 26-32, 34-45 and 47-49 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Ganesh et al. (U.S. Patent 6,957,236), hereafter "Ganesh" in view of Odom et al. (U.S. Patent 6,516,320), hereafter "Odom", and further in view of Najork et al. (U.S. Patent 7,007,027), hereafter "Najork". Claims 33 and 46 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Ganesh in view of Odom, and further in view of Najork, and further in view of Chang et al. (U.S. 6,584,459), hereafter "Chang". **Was a proper rejection made under 35 U.S. C. § 103(a) using existing USPTO guidelines**?

ARGUMENT:

Claims 26-32, 34-45 and 47-49 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Ganesh et al. (U.S. Patent 6,957,236), hereafter "Ganesh" in view of Odom et al. (U.S. Patent 6,516,320), hereafter "Odom", and further in view of Najork et al. (U.S. Patent 7,007,027), hereafter "Najork". Claims 33 and 46 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Ganesh in view of Odom, and further in view of Najork, and further in view of Chang et al. (U.S. 6,584,459), hereafter "Chang". **Was a proper rejection made under 35 U.S. C. § 103(a) using existing USPTO guidelines**?

Claims 26-32, 34-45 and 47-49 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Ganesh et al. (US 6,957,236), hereafter Ganesh, in view of Odom et al. (US 6,516,320), hereafter Odom, and further in view of Najork et al. (US 7,007,027), hereafter Najork. To be properly rejected under 35 U.S.C. § 103(a), the cited references have to provide for each and every element of the rejected claims. Applicants respectfully submit that the combination of the Ganesh, Odom and Najork fail to teach or suggest many of the features of the rejected claims.

Ganesh teaches a computer-implemented method for providing a data item to a transaction, wherein the method comprises the steps of: (a) locating, within volatile memory, a first version of a data block that includes a first version of the data item; (b) determining whether the first version of the data item is usable by the transaction without respect to whether the first

version of the data block is useable by the transaction; (c) if the first version of the data item is usable by the transaction, then establishing said data item as a candidate that can be provided to the transaction; and (d) if the first version of the data item is not useable by the transaction, then obtaining a version of the data item that is usable by the transaction from a second version of the data block that is different from said first version.

Odom teaches tiered hashing for data access wherein a memory for access by a program being executed includes a data access structure stored in memory, the data access structure including a first and second index structure together forming a tiered index. According to Odom, at least one entry in the first structure indicates an entry in the second structure.

Najork teaches a database comprising a B-Tree data structure having a plurality of nodes associated with disk blocks and handles stored in said nodes and a mechanism for performing a lookup operation with respect to a key, k, wherein, in traversing the B-Tree at a given node, the lookup operation refers to a left-link handle, h_{left}, of that node to access a left sibling of the node if the key k is less than or equal to a value K_{min} stored in the node.

Claim 26, by contrast, teaches a computer-based method to version a node range and locate a versioned node range in a storage architecture managing node ranges, said computer-based method implemented in computer readable program code stored in computer memory, said computer-based method comprising the steps of: (a) receiving a node modification request for a node range from a database system; (b) versioning said node range by copying, to a storage, a

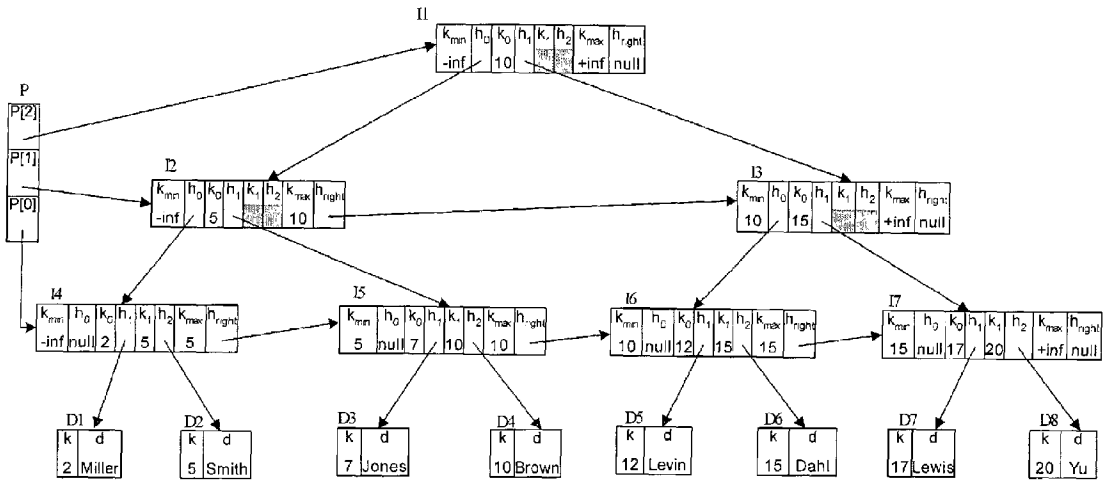
node range to which said node modification request is to be made and labeling said copied node range with an identifier; (c) locating said labeled node range via said identifier and a hash on said node range; and (d) outputting said located labeled node range.

With respect to independent claim 26, Applicants agree with the Examiner's conclusion on page 4 of the Office Action of 08/20/2007 that Ganesh and Odom fail to disclose, explicitly or implicitly, a storage architecture managing ***node ranges***. However, Applicants respectfully disagree with the Examiner's assertion on page 4 of the Office Action of 08/20/2007 that such node ranges are taught for by Najork. In the 'Response to Arguments' section of the Final Office Action of 03/03/2008, the Examiner erroneously reiterates this conclusion. For support of such an assertion, the Examiner points to column 2, lines 28-42 of Najork as teaching such a feature. The relevant citation from the Examiner's citation is provided below:

> "A mechanism for performing a lookup operation with respect to a key, k, traverses the B-Tree and refers to the left-link handle, h_{left}, of a node to access a left sibling of the node if the key k is less than or equal to a value k_{min} stored in the node. Mechanisms are also provided for performing insert and delete operations, and the lookup, insert, and delete operations detect if the ***key range of an index node***, A, does not-include the key k that the operation is trying to locate, and follow a handle $A.h_{left}$ to the left sibling when $k \leq A.k_{min}$." (emphasis added).

It is clear from the Examiner's own citation that **the key range is associated with an index node**. However, there appears to be no explicit or implicit mention of a "**node ranges**".

For further support, the Board of Patent Appeals and Interferences is respectfully directed to FIG. 3C of Najork et al., which is reproduced below:



Najork's FIG. 3C

It can be seen from the above-figure that **Najork's keys are not associated with node ranges, but rather map an "association" between a key, such as an employee number, to "values", such as employee records**. Najork in column 6, lines 39-41, specifically define this "association" in terms of a B-Tree data structure that maintains an association between such "keys" and "values". The above-reproduced figure from Najork shows this association. Specifically, **key 'k'=2 is associated with employee 'Miller', key 'k'=5 is associated with employee 'Smith', key 'k'=7 is associated with employee 'Jones', key 'k'=10 is associated with employee 'Brown', key 'k'=12 is associated with employee 'Levin', key 'k'=15 is associated with employee 'Dahl', key 'k'=17 is associated with employee 'Lewis', and key**

'k'=20 is associated with employee 'Yu'.

It should be clear from the above-presented discussion and the **keys and the key ranges of Najork are not associated with ranges associated with nodes, but are associated with 'k', which in turn is associated with an employee's number**. Hence, Applicants respectfully assert that a "**key range**" cannot be equated to a "**node range**". It should be noted that the Examiner in the 'Response to Arguments' section of the Final Office Action of 03/03/2008 correctly interprets Applicants' ranges of nodes as "**a set of nodes**" that "**form a range of node IDs called a node range**". However, the Examiner has failed to show such a node range in Najork. As outlined above, **Najork's range of keys are NOT associated with a range of node IDs, but are associated with a range of keys that are associated with employee numbers (and NOT associated with a range of nodes IDs)**.

Since Ganesh, Odom, or Najork fail to teach or suggest the management of **node ranges**, it also follows that the same combination of references also fail to teach or suggest **the versioning said node range by copying, to a storage, a node range to which said node modification request is to be made and labeling said copied node range with an identifier**.

Further, since Ganesh, Odom, or Najork fail to teach or suggest the management of **node ranges**, it also follows that the same combination of references also fail to teach or suggest the step of **locating a labeled node range via an identifier**, such as a timestamp or log sequence number, **and a hash on a node range**.

Absent such teachings, Ganesh, Odom, and Najork cannot render obvious Applicants' independent claim 26. Applicants, therefore, respectfully submit that an improper 35 U.S.C. §103 rejection was issued with regards to pending claim 26.

According to pending **claim 34**, the present invention provides for a computer-based method to version a node range and to locate a versioned node range in a storage architecture **managing node ranges via a node id range index, said each node assigned a node id value and a set of nodes forming a node range, each entry in said node id range index pointing to a node range and its range identifier, RID**, wherein the computer-based method is implemented in computer readable program code stored in computer memory, and the method comprises the steps of: (a) receiving a node modification request for a range; (b) versioning said range associated with said node modification request by shadowing nodes in said range to a Version Hash Table based on RID and assigning a time identifier to copies of said range; (c) locating a node in said shadowed range via said time identifier and RIDs; and (d) outputting said located node range.

With respect to independent claim 34, Applicants agree with the Examiner's conclusion on page 7 of the Office Action of 08/20/2007 that Ganesh and Odom fail to disclose, explicitly or implicitly, a storage architecture managing ***node ranges***. However, Applicants respectfully disagree with the Examiner's assertion on page 7 of the Office Action of 08/20/2007 that such node ranges are taught for by Najork. As mentioned in the argument of independent claim 26,

Applicants respectfully note that the Examiner's citation and the entire Najork reference merely teaches a "**key range**" and makes no explicit or implicit mention either a "**node range**" or an associated "**range identifier, RID**". Further, Applicants submit that the mere mention of a key range in no way anticipates or renders obvious the feature of **versioning by shadowing nodes in a range to a version has table based on RID**.

Since Ganesh, Odom, or Najork fail to teach or suggest a "**node range**" or a "**range identifier, RID**", it also follows that the same combination of references also fail to teach or suggest independent claim 34's feature of versioning a **range associated with said node modification request** by **shadowing nodes in said range** to a **Version Hash Table** based on **RID** and assigning a **time identifier** to copies of said range, wherein the **shadowed range** is located via said **time identifier and RIDs**;

Since Ganesh, Odom, or Najork fail to teach or suggest a "**node range**" or a "**range identifier, RID**", it also follows that the same combination of references also fail to teach or suggest independent claim 34's feature of **locating a node in a shadowed range via a time identifier and RIDs**.

Further, with respect to pending claim 34's feature of "shadowing nodes in said range to a Version Hash Table based on RID", the Examiner cites column 4, lines 61-65 of the Ganesh reference as teaching such a feature. This citation merely describes a situation **wherein updates are made to a copy of a data block to create a new version**. Conspicuously absent in the

above citation is a teaching or suggestion for ***node ranges*** or a teaching or suggestion for ***shadowing nodes*** in said range to a ***Version Hash Table based on RID***. Applicants maintain that there is neither an explicit nor an implicit mention in Ganesh for shadowing nodes or shadowing nodes in a node range to a table based on RID. Applicants, therefore, respectfully contend that the Ganesh reference fails to teach or suggest claim 34's features as asserted by the Examiner.

Further, with respect to pending claim 34's feature of "**assigning a time identifier to copies of said range**" and "**locating a node in said shadowed range via said time identifier and RIDs**", the Examiner cites column 4, lines 41-65 and column 2, lines 50-62 of the Ganesh reference as teaching such a feature. Column 4, lines 41-65 of the Ganesh merely addresses time parameters such as **INCLUDE TIME** and **EXCLUDE TIME**. **INCLUDE TIME specifies the commit time of the most recently committed transaction whose changes are included in the version of the data block**. **EXCLUDE TIME specifies the time at which the contents of the data block were "current"**. These definitions do not allow for such parameters of time to be associated with time identifiers that are assigned to copies of a node range, wherein such parameters can be used, along with Range IDs, to locate a shadowed range.

Applicants, therefore, respectfully contend that the Ganesh reference fails to teach or suggest claim 34's features as asserted by the Examiner.

Absent such teachings, Ganesh, Odom, and Najork cannot render obvious Applicants'

independent claim 34. Applicants, therefore, submit that an improper 35 U.S.C. § 103(a) rejection was issued with respect to claim 34.

Arguments presented above with respect to independent claims 26 and 34 substantially apply to independent claims 48 and 49. At least for the reasons presented above, Applicants respectfully assert that Ganesh, Odom, and Najork cannot render obvious Applicants' independent claims 48 and 49. Therefore, at least for the reasons set forth above, Applicants respectfully submit that an improper 35 U.S.C. § 103(a) rejection was issued with respect to claims 27-32 and 35-45.

The above-presented arguments with regards to independent claims 26 and 34 substantially apply to dependent claims 27-32 and 35-45 as they inherit all the features of the claim from which they depend. Therefore, at least for the reasons set forth above, Applicants respectfully submit that an improper 35 U.S.C. § 103(a) rejection was issued with respect to independent claims 48 and 49.

Claims 33 and 46 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Ganesh in view of Odom and further in view of Najork and further in view of Chang et al. (US 6,584,459), hereafter Chang. To be properly rejected under 35 U.S.C. §103(a), the cited references have to provide for each and every element of the rejected claims. Applicants respectfully submit that the combination of Ganesh, Odom, Najork and Chang fail to teach or suggest many of the features of claims 33 and 46. Therefore, at least for the reasons set forth

above, Applicants respectfully submit that an improper 35 U.S.C. § 103(a) rejection was issued with respect to claims 33 and 46.

Chang merely teaches an extender for a relational database system that includes indexing of structured documents with general and rich data types, querying structured documents using a conditional select function; and creating structure indexes using a tag counting system.

However, Chang fails to remedy the shortcomings of the Ganesh, Odom, and Najork references. Specifically, since Ganesh, Odom, or Najork fail to teach or suggest a "node range", it also follows that the same combination of references also fails to teach or suggest such **node ranges being associated with hierarchical node data derived from an XML document**.

Therefore, at least for the reasons set forth above, Applicants respectfully assert that the combination of Ganesh, Odom, Najork and Chang cannot render obvious the features of dependent claims 33 and 46. Applicants, therefore, respectfully submit that the Examiner issued an improper 35 U.S.C §103 rejection with regards to pending dependent claims 33 and 46.

SUMMARY

As has been detailed above, none of the references, cited or applied, provide for the specific claimed details of applicant's presently claimed invention, nor render them obvious. It is believed that this case is in condition for allowance and reconsideration thereof and early issuance is respectfully requested.

As this Appeal Brief has been timely filed within the set period of response, no fee for extension of time is required. However, the Commissioner is hereby authorized to charge any deficiencies in the fees provided, including extension of time, to Deposit Account No. 09-0460.

Respectfully submitted by
Applicant's Representative,

/ramraj soundararajan/

Ramraj Soundararajan
Reg. No. 53832

IP AUTHORITY, LLC
4821A Eisenhower Ave
Alexandria, VA 22304
(703) 461-7060

July 9, 2008

Claims Appendix:

Claims 1-25 (previously cancelled)

26. (Original) A computer-based method to version a node rage and locate a versioned node range in a storage architecture managing node ranges, said computer-based method implemented in computer readable program code stored in computer memory, said computer-based method comprising the steps of:

a. receiving a node modification request for a node range from a database system;

b. versioning said node range by copying, to a storage, a node range to which said node modification request is to be made and labeling said copied node range with an identifier;

c. locating said labeled node range via said identifier and a hash on said node range; and

d. outputting said located labeled node range.

27. (Previously Presented) The computer-based method of claim 26, wherein said identifier is any of the following: a timestamp or a LSN.

28. (Previously Presented) The computer-based method of claim 26, wherein said storage is a transient storage.

29. (Previously Presented) The computer-based method of claim 26, wherein said node modification request is any of the following: a node insertion request, a node update request, or a node deletion request.

30. (Previously Presented) The computer-based method of claim 26, wherein said method is implemented across a network.

31. (Previously Presented) The computer-based method of claim 30, wherein said network is any of the following: a local area network, a wide area network, or the Internet.

32. (Previously Presented) The computer-based method of claim 26, wherein said node ranges are associated with hierarchical node data that is derived from any of: a structured document, a computer network, or a directory file system.

33. (Original) The computer-based method of claim 32, wherein said structured document is an XML document.

34. (Original) A computer-based method to version a node range and to locate a versioned node range in a storage architecture managing node ranges via a node id range index, said each node assigned a node id value and a set of nodes forming a node range, each entry in said node id range index pointing to a node range and its range identifier, RID, said computer-based method implemented in computer readable program code stored in computer memory, said method comprising the steps of:

a. receiving a node modification request for a range;

b. versioning said range associated with said node modification request by shadowing nodes in said range to a Version Hash Table based on RID and assigning a time identifier to copies of said range;

c. locating a node in said shadowed range via said time identifier and RIDs; and

d. outputting said located node range.

35. (Original) The computer-based method of claim 34, wherein said time identifier is any of the following: timestamp or LSN.

36. (Original) The computer-based method of claim 34, wherein new readers, after a modification, access current nodes through a new RID.

37. (Original) The computer-based method of claim 34, wherein previous readers access old nodes via the same RID and hashing the same RID to locate the shadowed copy in said Version Hash Table.

38. (Original) The computer-based method of claim 34, wherein when modifications cause nodes in a range to be moved to a new RID, previous readers are redirected from the new RID to an old RID via a Redirection Hash Table.

39. (Original) The computer-based method of claim 34, wherein when modifications cause nodes in a range to be moved to a new RID, previous readers are redirected from the new RID to an old RID via an index that describes where old versions are in said Version Hash Table.

40. (Original) The computer-based method of claim 34, wherein said shadowed nodes are copied to a transient storage.

41. (Original) The computer-based method of claim 34, wherein said method is implemented across a network.

42. (Original) The computer-based method of claim 41, wherein said network is any of the following: a local area network, a wide area network, or the Internet.

43. (Original) The computer-based method of claim 34, wherein, for range deletions, the range being deleted is moved to reserved RID RIDFF.

44. (Original) The computer-based method of claim 43, wherein a reader hashes a Redirection Hash Table on RIDFF to find a correct Version Hash Table entry.

45. (Original) The computer-based method of claim 34, wherein said node ranges are associated with hierarchical node data that is derived from any of: a structured document, a computer network, or a directory file system.

46. (Original) The computer-based method of claim 45, wherein said structured document is an XML document.

47. (Original) The computer-based method of claim 34, wherein said node modification request is any of the following: a node insertion request, a node update request, or a node deletion request.

48. (Original) An article of manufacture comprising computer readable program code implementing a method to version a node range and to locate said versioned node in a storage architecture that manages node ranges via a node id range index, said each node assigned a node id value and a set of nodes forming a node range, each entry in said node id range index pointing to a node range and its range identifier, RID, said method comprising:

a. computer readable program code identifying a node modification request for a range;

b. computer readable program code versioning said range associated with said node modification request by shadowing nodes in said range to a Version Hash Table based on RID and assigning a time identifier to copies of said range;

c. computer readable program code locating a node in said shadowed range via said time identifier and RIDs; and

d. computer readable program code outputting said located node range.

49. (Original) An article of manufacture comprising computer readable program code implementing a method to version a node range and to locate a versioned node rage in a storage architecture that manages node ranges, said method comprising:

a. computer readable program code identifying a request for node modification from a database system;

b. computer readable program code copying, to a storage, a node range to which said node modification request is to be made;

c. computer readable program code labeling said copied node range with an identifier; and

d. computer readable program code locating said labeled node range via said identifier and a hash on said node range; and

e. computer readable program code outputting said located labeled node range.

Evidence Appendix

None

Related Proceedings Appendix

None